

Code the functions listed below and use the specified test cases. If you cheat on this assignment, you will most likely do poorly with the LISP coding on the final exam.

Notes:

- Look at the set up information for more information on executing LISP files.
- The **only** functions you can use are either those we discussed in the LISP notes (including ones we developed as exercises) or you can reuse any of the functions developed below in subsequent functions.
- Place your code in p2Lisp.txt.
- Load your code using (load "p2Lisp.txt" :echo T :print T).
- To execute the test cases using the file I provided:
(load "p2LispRun.txt" :echo T :print T)
- Your functions must be executed on a **fox** server using the specified test cases.
- Turn in a zip file named LastNameFirstName.zip(no spaces) containing:
 - Your source LISP code (p2Lisp.txt)
 - Your log of the session. Select all the text in the terminal window and paste it into a file named p2Out.txt
 - Do **not** have any directories within your zip file.
- In BlackBoard, include a note that specifies whether you did the extra credit.
- Your code must follow my **LISP programming standards** which are on my website for this course.

1. Code the function, removeNILTop, which is passed a list and removes NIL at the top level.

Examples:

```
> (removeNILTop '(NIL X NIL NIL Y NIL Z))  
  (X Y Z)  
> (removeNILTop '(X NIL Y NIL Z NIL))  
  (X Y Z)  
> (removeNILTop '(NIL (X NIL Y) (NIL NIL)))  
  ((X NIL Y) (NIL NIL))
```

2. Code the function, removeNILMost, which is passed a list and removes NIL at any level. Note: if the result of removing NIL gives a NIL, it is unnecessary to remove that resulting NIL. (See the extra credit.)

Examples:

```
> (removeNILMost '(NIL X NIL NIL Y NIL Z))  
  (X Y Z)  
> (removeNILMost '(X NIL (Y NIL Z) NIL))  
  (X (Y Z))  
> (removeNILMost '(NIL (NIL) (X NIL Y) (NIL NIL) Z))  
  (NIL (X Y) NIL Z)  
> (removeNILMost '(NIL ( (((((NIL) NIL)))))))
```

3. Code the function, reverseTop, which is passed a list and returns a reversed list of the high-level entries.

Hint: APPEND could be useful.

Examples:

```
> (reverseTop '(X Y Z))  
  (Z Y X)  
> (reverseTop '(X (Y Z (A)) (W)))  
  ((W) (Y Z (A)) X)
```

4. Code the function, reverseAll, which is passed a list and returns a reversed list at all levels.

Examples:

```
> (reverseAll '(X Y Z))  
  (Z Y X)  
> (reverseAll '(X (Y Z (A)) (W)))  
  ((W) ((A) Z Y) X)
```

(((((NIL))))))

5. Code the function, `palindrome`, which is passed a list and returns `T` if the list is a palindrome; otherwise, it returns `NIL`. It only needs to be a palindrome at the top-level.

Examples:

```
> (palindrome '(R A C E C A R))
```

`T`

```
> (palindrome '(W A S I T A C A R O R A C A T I S A W))
```

`T`

```
> (palindrome '(N I X O N))
```

`NIL`

Extra credit: (2 pts + 120 / n where n is the number of people (from all sections) that get it completely right)

Code the function `removeNILAll` which also removes any resulting `NIL` (except the single outermost)

```
> (removeNILAll '(NIL (NIL) (X NIL Y) (NIL NIL) Z))
```

`((X Y) Z)`

```
> (removeNILAll '(NIL ( (((((NIL) NIL)))))))
```

`NIL`

To receive extra credit:

- `removeNILAll` must work for all possible cases.
- The other functions must work for all possible cases.
- **All your functions** must be properly **documented**.
- Your submission **MUST NOT be late**.
- There is an extra file to run your code that includes extra credit cases named "p2LispExtraRun.txt"